第27卷 第2期　　工 程 数 学 学 报　　Vol. 27 No. 2
2010年04月　　CHINESE JOURNAL OF ENGINEERING MATHEMATICS　　Apr. 2010

文章编号:1005-3085(2010)02-0233-09

非定常Stokes方程一种基于完全重叠型区域分解的有限元并行算法*

尚月强[1,2]，何银年[2]

(1- 贵州师范大学数学与计算机科学学院，贵阳 550001；2- 西安交通大学理学院，西安 710049)

摘 要: 基于完全重叠型区域分解技巧，本文提出了一种求解非定常Stokes方程的有限元并行算法。该算法的基本思想是首先对空间施行完全重叠型区域分解，然后各个处理器使用向后Euler格式独立并行求解关于时间t的常微分方程；在整个关于时间的迭代过程中，无需处理器间的通信，具有良好的并行性能。该算法中每个处理器所负责的子问题是一个全局问题，它定义在整个求解区域上，但绝大部分自由度来自其所负责的子区域，从而使得该算法稍加修改现有的串行程序即可实现相应的并行计算，实现简单，具有重要的使用价值。同时通过数值算例，在曙光集群并行机上编程实现了上述算法，验证了其有效性。

关键词: Stokes方程；有限元方法；重叠型区域分解；并行算法

分类号: **AMS**(2000) 65M15; 65M60; 76D07; 76M10　　**中图分类号**: O241.82　　**文献标识码**: A

1 引言

设Ω是$\mathbf{R}^d\,(d=2,3)$中具有Lipschitz连续边界的有界开集，我们考虑非定常Stokes方程

$$\begin{aligned}
&u_t-\nu\Delta u+\nabla p=f, && \text{在 } \Omega\times(0,T] \text{ 内},\\
&\operatorname{div} u=0, && \text{在 } \Omega\times(0,T] \text{ 内}\\
&u=0, && \text{在 } \partial\Omega\times(0,T] \text{ 上},\\
&u=u_0, && \text{在 } \Omega\times\{0\} \text{ 上},
\end{aligned}\tag{1}$$

其中$u=(u_1(x,t),\cdots,u_d(x,t))$为速度向量，$p=p(x,t)$为压力，$f=(f_1(x,t),\cdots,f_d(x,t))$为体积力，$\nu$是粘性系数，$u_0$是初始速度向量，$T$是一给定的有限时间，$u_t=\frac{\partial u}{\partial t}$。方程(1)是流体力学的基本方程—Navier-Stokes方程的线性化形式，描述了蠕动流的流动情况。因此，其并行数值求解方法的研究具有重要的现实意义。

区域分解算法是一类数值求解偏微分方程的有效方法，它基于“分而治之”的思想，将求解区域分解成若干个形状相对较规则的子区域，使得原方程转化为在子区域上求解，从而将大型问题分解为小型问题、复杂边值问题分解成简单边值问题、串行问题分解成并行问题，以达到扩大求解问题的规模、提高计算精度、加快计算速度的目的[1]。对于非定常问题，传统的区域分解算法是首先对时间进行离散，然后在每一时间步对空间施行区域分解实现并行计算。这

收稿日期: 2009-12-01.　**作者简介**: 尚月强(1976年12月生)，男，博士，副教授. 研究方向：偏微分方程数值解与并行计算.

***基金项目**: 国家自然科学基金(10971166)；国家重大基础研究项目(973计划)(2005CB321703)；贵州省科学技术基金([2008]2123).

类方法的一个最大缺点是处理器之间的通信较频繁，因而其并行性能往往不够理想。本文提出求解非定常Stokes方程(1)的一种新型区域分解有限元并行算法，该算法首先对空间施行完全重叠型区域分解，然后在每个处理器使用向后Euler格式独立并行求解关于时间t的常微分方程，在整个关于时间的迭代过程中，无需处理器间的通信，具有良好的并行性能。同时我们在曙光集群并行机上进行了数值试验，验证了其有效性。

2 预备知识

2.1 非定常Stokes方程解的存在唯一性

对于非负整数k，我们按通常意义定义Sobolev空间$H^k(\Omega)$[2]，并令$H_0^1(\Omega)=\{u\in H^1(\Omega): u|_{\partial\Omega}=0\}$，这里$u|_{\partial\Omega}=0$是在迹的意义下。本文中，我们用$(\cdot,\cdot)$表$L^2(\Omega)^d\,(d=1,2,3)$上的标准内积，用$\|v\|_{m,\Omega}$表$H^k(\Omega)(k\geq m)$空间中的$m$-全范。令

$$X=H_0^1(\Omega)^d,\quad Y=L^2(\Omega)^d,\quad M=L_0^2(\Omega)=\Big\{q\in L^2(\Omega):\int_\Omega q\mathrm{d}x=0\Big\},$$

并定义双线性形式$a(\cdot,\cdot)$, $d(\cdot,\cdot)$

$$a(u,v)=\nu(\nabla u,\nabla v),\quad d(v,q)=(\mathrm{div}\,v,q),\quad \forall\, u,v\in X,\quad q\in M, \tag{2}$$

我们可得方程(1)的变分形式为：求$(u,p)\in X\times M$，使得对任意给定的$t\in(0,T]$，有

$$\begin{aligned}&(u_t,v)+a(u,v)-d(v,p)+d(u,q)=(f,v),\quad \forall\,(v,q)\in X\times M.\\&u(0)=u_0.\end{aligned} \tag{3}$$

假设初值u_0及右端项f满足以下条件：

H0 $u_0\in(H^2(\Omega)\cap H_0^1(\Omega))^d$，$\mathrm{div}u_0=0$，$f$, $f_t\in L^2(0,T;L^2(\Omega)^d)$满足

$$\|u_0\|_2+\Big(\int_0^T\big(\|f\|_0^2+\|f_t\|_0^2\big)\mathrm{d}t\Big)^{1/2}\leq c. \tag{4}$$

本文中c是一大于零的常数，它在不同的地方可能代表不同的值。对于方程(3)，我们有以下结果。

定理2.1[3,4] 假设H0成立，Ω是$\mathbf{R}^d$中具有$C^{k+1}\,(k\geq 1)$型边界的有界区域或$\mathbf{R}^d$中的凸多边形或凸多面体区域$(k=1)$，则对于任意的$0<T<\infty$，方程(3)存在唯一解(u,p)满足

$$\sup_{0<t\leq T}\big(\|u(t)\|_2^2+\|p(t)\|_1^2+\|u_t(t)\|_0^2\big)\leq c, \tag{5}$$

$$\sup_{0<t\leq T}\sigma(t)\|u_t\|_1^2+\int_0^T\sigma(t)\big(\|u_t\|_2^2+\|p_t\|_1^2+\|u_{tt}\|_0^2\big)\mathrm{d}t\leq c, \tag{6}$$

其中$\sigma(t)=\min\{1,t\}$。

2.2 非定常Stokes方程的有限元逼近

设$T^h(\Omega)=\{K\}$是Ω的一个满足一般正则性条件[5-7]的尺度为h的网格剖分，$X_h^0(\Omega)\subset X$, $M_h^0(\Omega)\subset M$是相应于网格$T^h(\Omega)$的两个有限元空间，满足逼近性质：任给$(u,p)\in H^{k+1}(\Omega)^d\times H^k(\Omega)(k\geq 1)$，存在$(\pi_h u,\rho_h p)\in X_h^0(\Omega)\times M_h^0(\Omega)$，使得

$$\|u-\pi_h u\|_{1,\Omega}\leq ch^s\|u\|_{1+s,\Omega},\quad \|p-\rho_h p\|_{0,\Omega}\leq ch^s\|p\|_{s,\Omega},\quad 0\leq s\leq k, \tag{7}$$

和 inf-sup (LBB) 条件：存在 $\beta > 0$，使得

$$\beta\|q\|_{0,\Omega} \le \sup_{v\in X_h^0(\Omega),v\neq 0} \frac{(\operatorname{div} v, q)}{\|\nabla v\|_{0,\Omega}}, \quad \forall\, q \in M_h^0(\Omega). \tag{8}$$

使用向后 Euler 格式对时间进行离散，我们可得方程 (3) 的全离散有限元格式为：求 $(u_h^n, p_h^n) \in X_h^0(\Omega) \times M_h^0(\Omega)$，使得

$$\begin{aligned} &\tfrac{1}{\Delta t}(u_h^n - u_h^{n-1}, v) + a(u_h^n, v) - d(v, p_h^n) + d(u_h^n, q) = (f^n, v), \quad \forall\, (v,q) \in X_h^0(\Omega) \times M_h^0(\Omega),\\ &u_h^0 = u_{h0}, \end{aligned} \tag{9}$$

其中 Δt 为时间步长满足 $0 < \Delta t < 1$，$f^n = \frac{1}{\Delta t}\int_{t_{n-1}}^{t_n} f\mathrm{d}t$，$t_n = n\Delta t$，$n = 1,2,\cdots,N$，$N = \lceil T/\Delta t\rceil$，$u_{h0}$ 是 u_0 的一个有限元逼近。

对于方程 (9)，我们有以下结果。

定理 2.2[3,4]　在定理 2.1 的条件下，对所有的 $n = 1,2,\cdots,N$，方程 (9) 存在唯一解 $(u_h^n, p_h^n) \in X_h^0(\Omega) \times M_h^0(\Omega)$ 满足

$$\|\nabla(u(t_n) - u_h^n)\|_{0,\Omega} + \|p(t_n) - p_h^n\|_{0,\Omega} \le c(\Delta t + h^s), \quad 1 \le s \le k, \tag{10}$$

其中 $c = c(\sigma(t), \Omega, u_0, f, T)$。

3　非定常 Stokes 方程基于完全重叠型区域分解的有限元并行算法

3.1　完全重叠型区域分解技巧

首先将求解区域 Ω 分解成互不重叠的子区域 $D_1, D_2, \cdots, D_J$，然后在 Ω 内将每一子区域向外扩展一定尺度获得 $\Omega_1, \Omega_2, \cdots, \Omega_J$ ($\Omega_j \subset \Omega$, $J = 1,2,\cdots,J$)，这些子区域 $\Omega_1, \Omega_2, \cdots, \Omega_J$ 构成 Ω 的一个重叠型区域分解。每台处理器负责一个子区域，各自对所负责的子区域生成尺度为 h 的细网格 $T^h(\Omega_j)$，并对其余区域生成尺度为 $H \gg h$ 的粗网格 $T^H(\Omega\backslash\Omega_j)$，然后采用自适应处理等技巧使得粗细网格在交界处相容(即任两个单元的交集或者为空集，或者为一公共顶点，或者为一公共边或为一公共面)，记这样的整个区域的多尺度网格为 $T_j^{H,h}(\Omega)$ ($j = 1,2,\cdots,J$)。这样的网格也可通过对整个区域的初始粗网格 $T^H(\Omega)$ 施行局部加密并通过自适应过程使之相容得到。图 1 描绘了四个子区域情形的完全重叠区域分解和相应的网格剖分情况。在该技巧中，相应于每一子区域 Ω_j 的子问题定义在整个求解区域 Ω 上，但其绝大部分自由度来自该子区域 Ω_j；每一个子问题事实上是一个全局问题，因而稍加修改现有的串行程序即可实现并行计算，实现简单，通信需求少。

3.2　基于完全重叠型区域分解的有限元并行算法

设 $D_1, D_2, \cdots, D_J$ 是求解区域 Ω 的一个互不重叠的区域分解，$\Omega_j \subset \Omega$ ($j = 1,2,\cdots,J$) 通过扩展 D_j ($j = 1,2,\cdots,J$) 所得。$T_j^{H,h}(\Omega)$ ($j = 1,2,\cdots,J$) 是 Ω 的多尺度网格剖分，它在 Ω_j 内的尺度为 h，而在远离 Ω_j 的地方尺度为 H (参见图 1)，这些 $T_j^{H,h}(\Omega)$ ($j = 1,2,\cdots,J$) 构成了 Ω 的一个完全重叠型区域分解。设 $X_{H,h,j}^0(\Omega)$ 和 $M_{H,h,j}^0(\Omega)$ 是相应于 $T_j^{H,h}(\Omega)$ ($j = 1,2,\cdots,J$) 的有限元空间，则求解方程 (1) 基于上述完全重叠型区域分解的有限元并行算法如下：

算法 1　非定常 Stokes 方程基于完全重叠型区域分解的全离散有限元并行算法

对 $n = 1,2,\cdots,N$：

1) 并行求 $(u_{H,h}^{j,n}, p_{H,h}^{j,n}) \in X_{H,h,j}^0(\Omega) \times M_{H,h,j}^0(\Omega)\,(j=1,2,\cdots,J)$，使得

$$\Big(\frac{u_{H,h}^{j,n}-u_{H,h}^{j,n-1}}{\triangle t}, v\Big) + a(u_{H,h}^{j,n}, v) - d(v, p_{H,h}^{j,n}) + d(u_{H,h}^{j,n}, q) = (f^n, v),$$
$$\forall\,(v,q) \in X_{H,h,j}^0(\Omega) \times M_{H,h,j}^0(\Omega),$$

$$u_{H,h}^{j,0} = P_{H,h,j}u_0.$$

2) 在 D_j 内取 $(u_n^h, p_n^h) = (u_{H,h}^{j,n}, p_{H,h}^{j,n})\,(j=1,2,\cdots,J)$。其中 $P_{H,h,j}: Y \to X_{H,h,j}^0(\Omega)$ 是正交投影，定义为

$$(P_{H,h,j}u, v) = (u,v), \quad \forall\, u \in Y, \quad v \in X_{H,h,j}^0(\Omega), \quad j=1,2,\cdots,J.$$

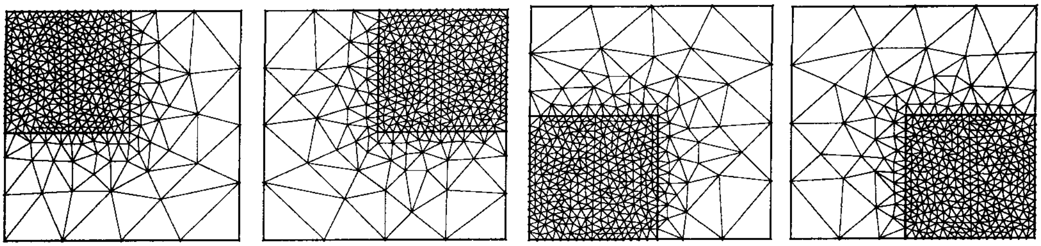


图 1: 四个子区域情形完全重叠型区域分解

3.3 算法分析

由算法1可知，新算法中每一子问题实际上是一个全局问题，从而稍加修改现有的串行程序即可实现并行计算，实现简单；在关于时间的迭代过程中，无需处理机之间的通信，具有较好的并行性能。定义分片范数

$$|||\nabla(u-u^h)|||_{0,\Omega} = \Big(\sum_{j=1}^{J} \|\nabla(u-u^h)\|_{0,D_j}^2\Big)^{1/2}, \quad |||p-p^h|||_{0,\Omega} = \Big(\sum_{j=1}^{J} \|p-p^h\|_{0,D_j}^2\Big)^{1/2}.$$

针对定常Stokes方程，基于两重网格离散技巧，何银年教授等人在文献[8]中提出了类似于完全重叠型区域分解的有限元并行算法，并在网格和有限元空间的一些假设条件(即：逼近性、反估计、超收敛性、LBB条件)下，得到下列误差估计

$$|||\nabla(u-u^h)|||_{0,\Omega} + |||p-p^h|||_{0,\Omega} \le c(h^s + H^{s+1}), \quad 1 \le s \le k. \tag{11}$$

对于非定常问题，由于与时间相关，其理论分析比定常问题要复杂得多。本文我们就所提出的新算法1，主要从数值模拟的角度考察其有效性及并行性能，其相关理论分析我们将在后续工作中给出。

4 数值试验

本节我们给出两个数值算例以验证算法的有效性。数值试验平台为西安交通大学理学院智能信息处理与计算实验室的曙光集群并行机，其每个结点具有8核2.0GHz CPU, 2GB × 8内存，各结点通过20Gbps InfiniBand连在一起。消息传递软件为MPICH。

算例1　解析解

针对求解区域$\Omega=[0,1]\times[0,1]$和已知解析解的情形，即准确解为

$$\begin{aligned} u_1 &= x^2(x-1)^2y(y-1)(2y-1)\cos(t),\\ u_2 &= -y^2(y-1)^2x(x-1)(2x-1)\cos(t),\\ p &= (3x^2+3y^2-2)\cos(t), \end{aligned} \tag{12}$$

我们设计数值试验如下：首先将求解区域分解成四个子区域，分别用P_1b-P_1元和文献[9]中所讨论的稳定化P_1-P_1和P_1-P_0元计算相应的有限元解，考察其关于h的渐进误差；然后考察算法1的并行性能。

P_1b-P_1元的有限元空间定义为

$$X^0_{H,h,j}(\Omega)=\left\{v\in H^1_0(\Omega)^2: v|_K\in(P_1\oplus\mathrm{span}\{b(K)\})^2,\ \forall\, K\in T_j^{H,h}(\Omega)\right\},\quad j=1,\cdots,J,$$

$$M^0_{H,h,j}(\Omega)=\left\{q\in L^2_0(\Omega)\cap C^0(\Omega): q|_K\in P_1,\ \forall\, K\in T_j^{H,h}(\Omega)\right\},\quad j=1,\cdots,J,$$

其中$b(K)$为单元K上次数为3的泡函数，满足$b(K)|_{\partial K}=0$，P_1为线性多项式函数空间。

对于稳定化的P_1-P_1元和P_1-P_0元，其速度空间定义为

$$X^0_{H,h,j}(\Omega)=\left\{v\in H^1_0(\Omega)^2: v|_K\in(P_1)^2,\ \forall\, K\in T_j^{H,h}(\Omega)\right\},\quad j=1,\cdots,J.$$

压力空间分别为

$$M^0_{H,h,j}(\Omega)=\left\{q\in L^2_0(\Omega)\cap C^0(\Omega): q|_K\in P_1,\ \forall\, K\in T_j^{H,h}(\Omega)\right\},\quad j=1,\cdots,J,$$

$$M^0_{H,h,j}(\Omega)=\left\{q\in L^2_0(\Omega): q|_K\in P_0,\ \forall\, K\in T_j^{H,h}(\Omega)\right\},\quad j=1,\cdots,J,$$

相应的稳定化格式为

$$\begin{aligned} &\left(\tfrac{u^{j,n}_{H,h}-u^{j,n-1}_{H,h}}{\Delta t},v\right)+a(u^{j,n}_{H,h},v)-d(v,p^{j,n}_{H,h})+d(u^{j,n}_{H,h},q)+C(p^{j,n}_{H,h},q)=(f^n,v),\\ &\qquad\qquad\forall\,(v,q)\in X^0_{H,h,j}(\Omega)\times M^0_{H,h,j}(\Omega), \end{aligned} \tag{13}$$

$$u^{j,0}_{H,h}=P_{H,h,j}u_0.$$

其中稳定项$C(\cdot,\cdot)$定义为

$$C(p,q)=\big((I-\Pi)p,(I-\Pi)q\big),\quad \forall\, p,\,q\in L^2(\Omega), \tag{14}$$

$\Pi: L^2(\Omega)\to P_0$(对P_1-P_1元)或$\Pi: L^2(\Omega)\to P_1$(对P_1-P_0元)为压力投影算子。

首先，我们将求解区域划分成四个互不重叠的子区域

$$D_1=(0,1/2)\times(0,1/2),\quad D_2=(1/2,1)\times(0,1/2),$$

$$D_3=(0,1/2)\times(1/2,1),\quad D_4=(1/2,1)\times(1/2,1),$$

然后在Ω内将每一子区域向外扩展尺度为H的区域获得$\Omega_j\,(j=1,2,3,4)$，并分别就$h=n^{-2}$，$H=\sqrt{h}\,(n=4,6,8,10)$进行计算，其中$\nu=1$，$\Delta t=0.001$。计算结果如表1所示。

表1: 算法近似解的误差: $\Delta t=0.001$，四子区域情形

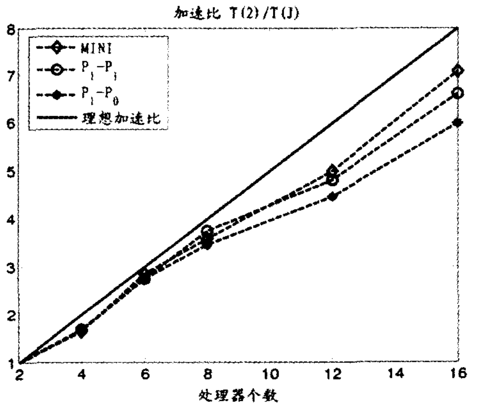
T	方法	h	H	$\frac{\|\|\nabla(u(t_n)-u_n^h)\|\|_{0,\Omega}}{\|\|\nabla u(t_n)\|\|_{0,\Omega}}$	$\frac{\|\|p(t_n)-p_n^h\|\|_{0,\Omega}}{\|\|p(t_n)\|\|_{0,\Omega}}$	u_{H^1} 收敛率	p_{L^2} 收敛率
$T=0.01$	P_1b-P_1	1/16	1/4	0.138366	0.00363449		
		1/36	1/6	0.0628475	0.00151349	0.97325	1.08036
		1/64	1/8	0.03549	0.00084807	0.99330	1.00678
		1/100	1/10	0.0224415	0.00053001	1.02712	1.05341
	P_1-P_1	1/16	1/4	0.230312	0.0184835		
		1/36	1/6	0.0965232	0.00631698	1.07247	1.32402
		1/64	1/8	0.0529758	0.00314746	1.04281	1.21087
		1/100	1/10	0.0347329	0.00193007	0.94602	1.09591
	P_1-P_0	1/16	1/4	0.2113740	0.04876540		
		1/36	1/6	0.0896269	0.02156560	1.05807	1.00622
		1/64	1/8	0.0499565	0.01208310	1.01597	1.00692
		1/100	1/10	0.0330756	0.00768306	0.92408	1.01468
$T=0.1$	P_1b-P_1	1/16	1/4	0.1416800	0.00339374		
		1/36	1/6	0.0659875	0.00140108	0.948467	1.09713
		1/64	1/8	0.0384007	0.00075508	0.949655	1.08312
		1/100	1/10	0.0248008	0.00045775	0.990862	1.13267
	P_1-P_1	1/16	1/4	0.2351050	0.01788110		
		1/36	1/6	0.0991256	0.00626389	1.07118	1.29968
		1/64	1/8	0.0555805	0.00316079	1.01425	1.19748
		1/100	1/10	0.0372184	0.00193785	0.90981	1.10747
	P_1-P_0	1/16	1/4	0.218438	0.0485246		
		1/36	1/6	0.0933633	0.0216298	1.05436	1.00256
		1/64	1/8	0.0539656	0.0121948	0.96140	1.00471
		1/100	1/10	0.0366619	0.0077892	0.87750	1.01567

计算结果表明：当 $T=0.01$ 时，三种有限元方法所得近似解关于网格参数 h 都取得了接近标准有限元方法的一阶收敛率；但随着时间向前推进，其速度的精度和收敛阶有所降低。在所得近似解精度方面，不论是速度还是压力，P_1b-P_1 元所得近似解的精度明显高于稳定化的 P_1-P_1 元和 P_1-P_0 元，这有别于标准的全局算法 (参见文献 [9] 的数值结果)；而稳定化的 P_1-P_1 元和 P_1-P_0 元之间，P_1-P_1 元所计算的压力精度显著高于 P_1-P_0 元，但其所得速度的精度却稍差于 P_1-P_0 元。

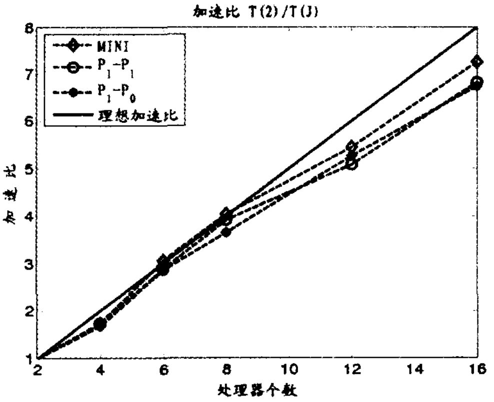
其次，我们将求解区域分别划分成2-16个子区域进行相应的计算，以考察算法的并行性能。表2及图2分别报告了三种不同的有限元方法在曙光集群并行机上的运行时间、加速比、并行效率和加速比随处理器数的变化情况，其中Ves为网格 $T_j^{H,h}(\Omega)\,(j=1,2,\cdots,J)$ 中三角单元顶点数的最小者，$h=1/100$, $H=1/10$, $\Delta t=0.001$。数值结果表明：算法取得了较好的并行性能，而且关于时间的迭代次数越多，并行性能越好；当 $T=0.1$ 时，P_1b-P_1 元在处理器个数为4、6时还取得了超线性的加速比和并行效率。

表 2:　并行程序的运行时间 $T(J)$ (秒)、加速比 $S_p = \frac{T(2)}{T(J)}$ 和并行效率 $E_p = \frac{2\times T(2)}{J\times T(J)}$

T	子区域数 J	Ves	P_1b-P_1			P_1-P_1			P_1-P_0		
			$T(J)$	S_p	E_p	$T(J)$	S_p	E_p	$T(J)$	S_p	E_p
0.01	2	7234	176.55	1.0	1.0	138.52	1.0	1.0	135.56	1.0	1.0
	4	4734	107.79	1.64	0.82	82.07	1.69	0.85	80.91	1.67	0.84
	6	3499	61.8	2.86	0.95	50.32	2.75	0.92	49.04	2.76	0.92
	8	2885	49.31	3.58	0.90	36.88	3.76	0.94	39.22	3.46	0.87
	12	2202	35.27	5.01	0.84	28.73	4.82	0.80	30.34	4.47	0.75
	16	1830	24.94	7.08	0.89	20.95	6.61	0.83	22.57	6.01	0.75
0.1	2	7234	1765.2	1.0	1.0	1356.38	1.0	1.0	1330.26	1.0	1.0
	4	4734	1019.87	1.73	0.87	788.05	1.72	0.86	802.61	1.66	0.83
	6	3499	579.12	3.05	1.02	470.58	2.88	0.96	464.63	2.86	0.95
	8	2885	436.5	4.04	1.01	346.59	3.91	0.98	364.01	3.65	0.91
	12	2202	323.53	5.46	0.91	267.11	5.08	0.85	252.3	5.27	0.88
	16	1830	243.57	7.25	0.91	199.01	6.82	0.85	197.39	6.74	0.84



(a) $T = 0.01$



(b) $T = 0.1$

图 2　$h = 1/100$, $H = 1/10$, $\triangle t = 0.001$ 时算法的加速比 $\frac{T(2)}{T(J)}$ 随处理器数的变化情况

算例 2　后台阶问题

本算例考虑后台阶问题，它是检验一个算法有效性的很好的基准问题之一，其求解区域及边界条件如图 3 所示，其中体积力 $f = 0$。我们将求解区域分解成五个互不重叠的子区域(见图 4)，然后将每一子区域向外扩展尺度为 H 的区域以获得相互重叠的区域分解，其中 $h = 1/8$, $H = 1/4$，所使用的有限元为二阶 Taylor-Hood 元。时间步长 $\triangle t = 0.01$，初始速度 $u^0 = 0$, $\nu = 1$。

图 5 描绘了流体达到稳定状态，即满足迭代终止条件

$$\frac{\|u_{H,h}^{j,n+1} - u_{H,h}^{j,n}\|_{0,\Omega}}{\|u_{H,h}^{j,n+1}\|_{0,\Omega}} \leq 10^{-5}, \quad j = 1, 2, \cdots, J, \tag{15}$$

时速度和压力的等值线。需注意的是图5中每一子图在$x = 0,4,8,12$处的垂直线是子区域$D_j\,(j=1,\cdots,5)$的人工边界线。该算例进一步验证了本文所提出的并行算法的有效性。

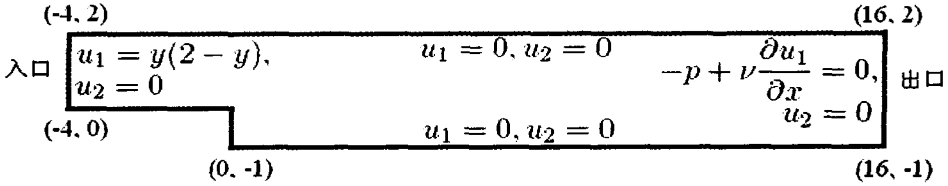


图 3: 后台阶问题

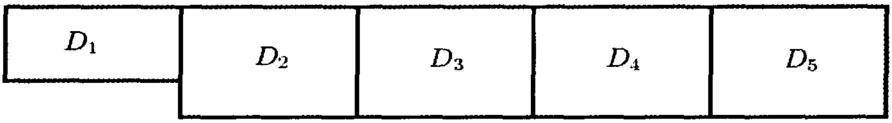
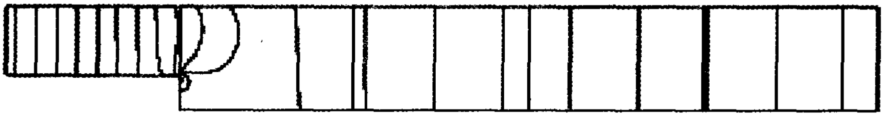


图 4: 后台阶问题的区域分解

(a) 速度 u_1

(b) 速度 u_2



(c) 压力 p

图 5: 台阶问题达到稳定状态 $(T = 0.92)$ 时近似解的等值线

5 结论

基于完全重叠型区域分解技巧，本文提出了一种新的求解非定常Stokes方程的有限元并行算法。该算法实现简单，可充分利用现有的串行软件进行并行编程，同时具有良好的并行性能。数值算例验证了其有效性。

致谢：西安交通大学理学院智能信息处理与计算实验室为本研究提供了并行数值计算平台；感谢靖稳峰老师给予的大力支持！

参考文献：

[1] Toselli A, Widlund O B. Domain Decomposition Methods: Algorithm and Theory[M]. Vol.34 of Springer Series in Computational Mathematics, Berlin: Springer, 2005

[2] Adams R. Sobolev Spaces[M]. New York: Academic Press Inc, 1975

[3] Heywood J G, Rannacher R. Finite element approximation of the nonstationary Navier-Stokes problem I: regularity of solutions and second-order error estimates for spatial discretization[J]. SIAM Journal on Numerical Analysis, 1982, 19(2): 275-311

[4] Girault V, Raviart P A. Finite Element Methods of the Navier-Stokes Equations—Theory and Algorithms[M]. Berlin: Springer-Verlag, 1986

[5] 李开泰，黄艾香，黄庆怀. 有限元方法及其应用[M]. 北京：科学出版社，2006
Li K T, Huang A X, Huang Q H. Finite Element Methods and Their Applications[M]. Beijing: Science Press, 2006

[6] 马逸尘，梅立泉，王阿霞. 偏微分方程现代数值方法[M]. 北京：科学出版社，2006
Ma Y C, Mei L Q, Wang A X. Modern Numerical Methods for Partial Differential Equations[M]. Beijing: Science Press, 2006

[7] Ciarlet P G, Lions J L. Handbook of Numerical Analysis, Vol.II, Finite Element Methods (Part I)[M]. Amsterdam: Elsevier Science Publisher, 1991

[8] He Y, Xu J, Zhou A, *et al.* Local and parallel finite element algorithms for the Stokes problem[J]. Numerische Mathematik, 2008, 109(3): 415-434

[9] Shang Y. New stabilized finite element method for time-dependent incompressible flow problems[J]. International Journal for Numerical Methods in Fluids, 2010, 62(2): 166-187

A Parallel Finite Element Algorithm Based on Fully Overlapping Domain Decomposition for the Time-dependent Stokes Equations

SHANG Yue-qiang[1,2], HE Yin-nian[2]

(1- School of Mathematics and Computer Science, Guizhou Normal University, Guiyang 550001;
2- Faculty of Science, Xi'an Jiaotong University, Xi'an 710049)

Abstract: Based on the fully overlapping domain decomposition, a parallel finite element algorithm for the time-dependent Stokes equations is proposed. The basic idea of the algorithm is to first discretize the spatial space by using the fully overlapping domain decomposition technique, then independently solve a system of ordinary differential equations with respect to time by the backward Euler scheme in overlapped subdomains. During the time-iterations, there is no communication between processors and hence a good parallel performance can be obtained. In this algorithm, each subproblem is a global problem with the vast majority of the degrees of freedom associated with the particular subdomain that it is responsible for, and hence can be solved with other subproblems in parallel by using an existing sequential solver without extensive recoding. Numerical tests on a Dawning parallel cluster illustrated the efficiency of the algorithm.

Keywords: Stokes equations; finite element method; overlapping domain decomposition; parallel algorithm

---

Received: 01 Dec 2009. **Accepted:** 16 Dec 2009.

Foundation item: The National Natural Science Foundation of China (10971166); the National Basic Research Program (973 Project)(2005CB321703); the Science and Technology Foundation of Guizhou Province ([2008]2123).